Un hôtel souhaite réaliser une application pour gérer la réservation de ses chambres. Une chambre est caractérisée par des équipements tels que la présence d'une douche ou/et d'une baignoire, d'un ou de plusieurs lits (simples ou doubles), etc. L'application doit gérer les réservations des clients (nom / prénom, date de début et de fin) et également sauvegarder l'horodatage des entrées des clients dans leur chambre.

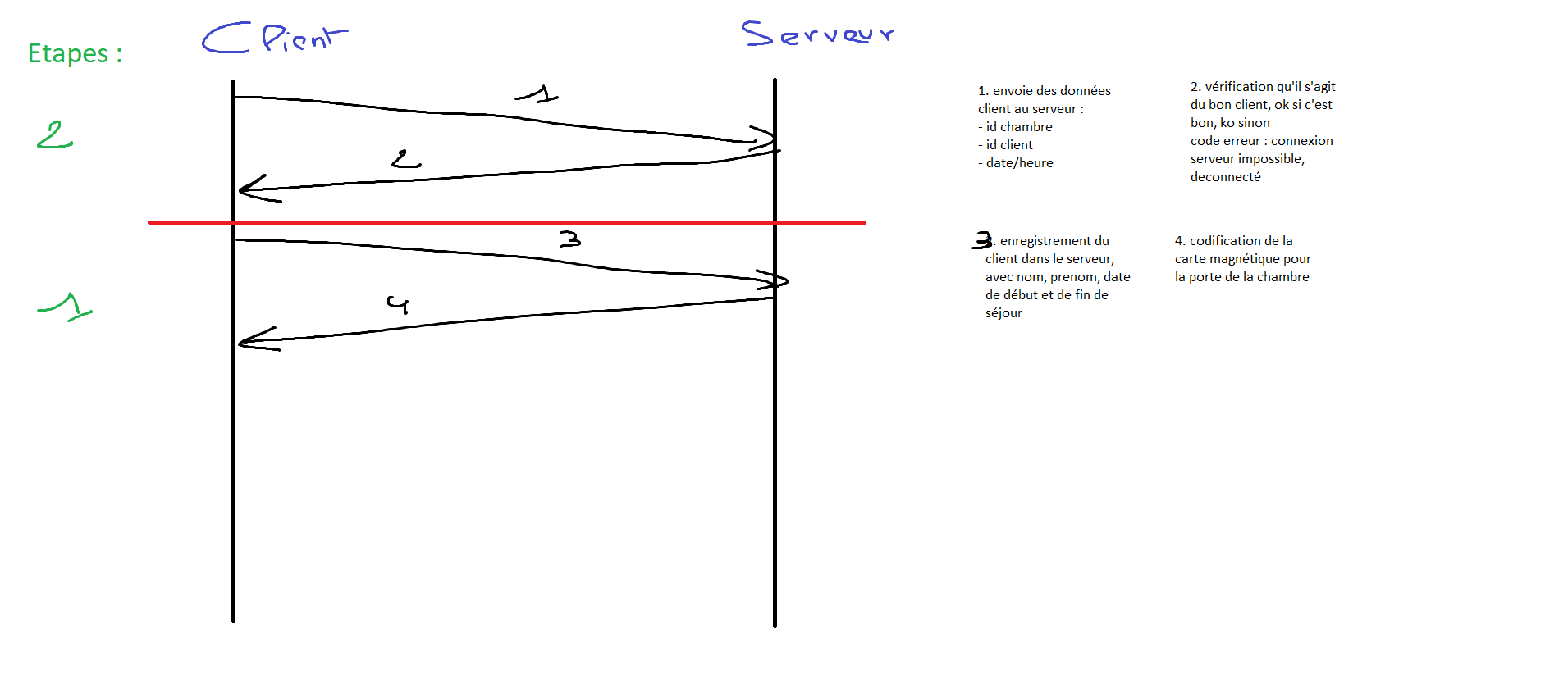
Chaque chambre possède un boitier individuel qui permet aux clients d'entrer dans leur chambre, à l'aide d'une carte magnétique. Chaque fois qu'un client entre dans sa chambre, le boitier envoie l'information au serveur central.

L'application possède les éléments suivants :

* Un serveur global qui stocke toutes les informations sur les chambres, ainsi que les logs de connexion
* Un client correspondant au boitier individuel pour chaque chambre

Les clients peuvent se connecter à un site Web pour réaliser les réservations.

**En utilisant *draw.io*, proposez une modélisation générale du système, ainsi que des différentes phases. Vous inclurez les images réalisées (exportées en jpg ou png) dans la zone de réponse.**



Une société privée souhaite développer une application afin de gérer son parking. Un parking est caractérisé par différents étages et à chaque étage, les places de parking sont numérotées. Certaines places peuvent être louées par des clients. Dans ce cas-là, les caractéristiques du véhicule sont stockées (immatriculation, modèle, marque) pour que les gardiens du parking puisse vérifier les véhicules présents. L'application doit également gérer l'ensemble des entrées/sorties.

L'application est constituée des éléments suivants :

* Le serveur central qui stocke toutes les informations sur le parking, mémorise les locations des clients (date de début et de fin, nom / prénom du client, caractéristiques du véhicule) et stocke les logs des entrées/sorties dans le parking
* Les lecteurs situés à chaque entrée du parking ; ils permettent de stocker chaque entrée de véhicule pour la facturation (immatriculation et heure) et chaque sortie (immatriculation, heure et tarif)
* Les lecteurs destinés aux gardiens qui leur permettent de vérifier que les véhicules sur les places louées sont les bons : ils peuvent saisir le numéro de place

Nous ne gérons pas ici les tarifs.

**Proposez des solutions techniques d'implémentation (protocoles, architecture...) permettant la mise en œuvre de l'architecture précédente. Vous préciserez le plan d'implantation en précisant les protocoles et technologies utilisées sur chaque acteur.**

serveur central :

- clients {place{etage, numero}, date debut, date fin, nom, prenom, caractéristique du véhicule{immatriculation, modele, marque}}

- log{immatriculation, heure}

lecteur entrées/sorties:

- véhicules {immatriculation}

- date

- tarif

lecteur gardiens :

- numéro place

si le client loue une place, une place lui est attribué le temps de sa location, si le client ne loue pas de place, une place non louée lui est donnée en entrant dans le parking.

les lecteurs d'entrees/sorties enregistres les véhicules, avec leurs immatriculation, via une caméra, ainsi que la date, le tarif est connu à l'avance. Ils envoient les informations d'entrées/sorties au serveur central.

le lecteur du gardien permet de saisir une place et d'avoir les caractéristiques du véhicule qui doit y être (immatriculation, modele, marque).

Une société privée souhaite développer une application afin de gérer son parking. Un parking est caractérisé par différents étages et à chaque étage, les places de parking sont numérotées. Certaines places peuvent être louées par des clients. Dans ce cas-là, les caractéristiques du véhicule sont stockées (immatriculation, modèle, marque) pour que les gardiens du parking puisse vérifier les véhicules présents. L'application doit également gérer l'ensemble des entrées/sorties.

L'application est constituée des éléments suivants :

* Le serveur central qui stocke toutes les informations sur le parking, mémorise les locations des clients (date de début et de fin, nom / prénom du client, caractéristiques du véhicule) et stocke les logs des entrées/sorties dans le parking
* Les lecteurs situés à chaque entrée du parking ; ils permettent de stocker chaque entrée de véhicule pour la facturation (immatriculation et heure) et chaque sortie (immatriculation, heure et tarif)
* Les lecteurs destinés aux gardiens qui leur permettent de vérifier que les véhicules sur les places louées sont les bons : ils peuvent saisir le numéro de place

Nous ne gérons pas ici les tarifs.

**L'ensemble des données du système seront sauvegardées et transiteront au format JSON. Listez et typez l'ensemble des données, ainsi que leurs localisations. Proposez une représentation JSON des données, et expliquez les différents flux associés.**

le serveur central stock les données sous le format JSON :

"Client"[

"place" {"etage", "numero"},

"date debut" {date, heure},

"date fin" {date, heure],

"nom", "prenom"

Vehicule{"immatriculation","modele", "marque"}

]

"VehiculeTemp"[

"immatriculation", "date entree" {date, heure}

]

"Log" [

"véhicule"{immatriculation, modele, marque},

"date"{date, heure},

"client"{boolean} //ceci est pour signifier si il s'agit d'un client enregistré ou non

]

les lecteurs (entrée/sortie) envoie les données au serveur central via la forme JSON :

{"immatriculation", "date"{date, heure}

les lecteurs des gardiens envoie la numérotation de etage/place :

{"etage", "place"}

et recois les informations :

{"occupation", "marque", "modele", "immatriculation"}

Pour assurer l’identification des personnes arrivant sur un campus universitaire, nous souhaitons mettre en place une application distribuée. Nous distinguons deux types de personnes : les « gardiens » et les « simples usagers ». Les simples usagers peuvent générer des attestations qui sont caractérisées entre autre, par une durée de validité (date de début et de fin) et un motif (cours, examen, etc.). Les gardiens peuvent accéder à toutes les attestations, ce qui leur permet de contrôler les usagers aux entrées du campus.  
  
L’application est constituée des éléments suivants :

* Un serveur d’authentification : il est permet de vérifier l’authentification d’un utilisateur ; il est en mesure également d’associer les rôles « gardien » ou « simple usager »
* Un serveur d’attestations : il stocke et permet de consulter/vérifier les attestations des usagers ; il est en mesure également de conserver un log de toutes les vérifications des attestations (une date) qui sont associées aux gardiens qui les ont effectué
* Un client pour les usagers: il permet aux usagers de créer des attestations qui sont stockées ensuite sur le smartphone ; elles possèdent un QRCode (une image)
* Un client pour les gardiens : il permet de numériser un QRCode pour vérifier une attestation

Les simples usagers peuvent générer leurs attestations via un site Web.

**On souhaite sécuriser cette application. Donnez les éléments qui vous semblent nécessaires à sécuriser, puis présentez les moyens utilisés. Enfin précisez l'ensemble des outils ainsi que, le cas échéant, les nouveaux acteurs qu'il faut inclure dans le système.**

les éléments à sécuriser sont : le serveur d'authentification, serveur d'attestation, les clients usagers et gardiens

les moyens à utiliser correspondant : une clé de chiffrements (RSA ou AES), des clés privées et publiques, RSA ou AES